数理解析研究所講究録
423巻 1981年 40-60

多層論理式からのデータベース・アクセス手順の生成について

東大 宇宙研　　宇田川 佳久
大須賀 節雄

1. はじめに

本稿は many Sorted Logic の記述力を拡張した多層論理について述べ，多層論理式によって表現されたデータベースの検索条件から検索手順を生成するアルゴリズムについて論じたものである。述語論理は知識表現の手段としても，またアルゴリズムの記述言語としても優れた特徴を備えている形式言語である。その中で many Sorted Logic は一階述語論理の問題点のいくつかを解決し，効率的な推論処理が行なえる特徴をもっている。しかし，many Sorted Logic が陽に含むことができるのは要素とそれが属する集合の関係であるために，集合をも変数として含むような記述をこの体系の範囲で行なうことは難しい。数多くの応用における実際的な知識を記述しようとするとき，値として要素をとる変数だけでなく集合又は巾集合などを値とする変数を含めなければならない。例えば，リレーショナル・データベース

への質問に限ってみても集合の要素の数を数えるCOUNT，集合の要素の値の平均値を求めるAVERAGE，集合の要素の最大値，最小値を求めるMAX，MINなどである。このようなMany Sorted Logicの制約を取り除き，しかも推論処理が行なえる形式言語として我々は多層論理を提案した。[11] Many Sorted Logicが全ての要素にSortを付随させているのに対して，多層論理式では全ての要素にSort又はSortの集合などの領域を割り当てることができ，要素のみならず集合や巾集合などを値とする変数を含む知識をこの体系の範囲で記述することができる。

本稿第2章では多層論理式の定義を与え，基本的な定理を証明する。これによって多層論理とMany Sorted Logicとの相違が明らかにされる．第3章では，リレーショナル・データベースの検索に関する知識と質問とを多層論理式によって記述する方法を例を用いながら論ずる．次に質問が推論処理によって基本的な手続きとデータベースを用いることによって評価できる論理式に変換されることを示す．第4章では推論処理の結果得られた多層論理式を満たす値を求める手続きを生成するアルゴリズムを示す．

2．多層論理式の構文，意味並びに基本的な定理

2．1　多層論理式の構文

多層論理式は以下に示す構文規則を満す言語である．

項：(1)定数は項である。(2) t を定数から成る項であるとき t の集合 {t} は項である。(3)変数は項である。なお，必要に応じて定数を値とする変数を要素変数，定数の集合を値とする変数を集合変数等と呼ぶことがある。

限量子：D を集合とするとき *D で D の巾集合を表わすとする．また，変数 $x_0, \cdots, x_n$ の領域が $Dom\{x_0\} \in \cdots \in Dom\{x_n\} = *_n \cdots *_1 D$ なる関係を満たすとする。このとき $(Q_n x_n / *_n \cdots *_1 D)(Q_{n-1} x_{n-1} / x_n) \cdots (Q_0 x_0 / x_1)$ を限量子と呼ぶ。ただし，Q_i $(0 \leqq i \leqq n)$ は量記号 ∀ 又は ∃ のいずれかである．$n = 0$ の場合，限量子は $(Q_0 x_0 / D)$ でありMany Sorted Logic の限量子と等しい。

アトム：G を n 引数述語記号とし，$t_1, \cdots, t_n$ を項とするとき $G(t_1, \cdots, t_n)$ はアトムである $(n \geqq 0)$。

多層論理式：(1)アトムは多層論理式である。(2) F と G が多層論理式ならば，$\sim F$，$F \vee G$，$F \wedge G$，$F \rightarrow G$，$F \leftrightarrow G$ は多層論理式である．(3) G が変数 $x_0, x_1, \cdots, x_n$ を自由変数として含む多層論理式であり，ある集合 D に対しこれらの変数の領域が $Dom\{x_0\} \in Dom\{x_1\} \in \cdots \in Dom\{x_n\} = *_n \cdots *_1 D$ なる関係を満しているとき $(Q_n x_n / *_n \cdots *_1 D)(Q_{n-1} x_{n-1} / x_n)$

$\cdots(Q_0 x_0 / x_1)\, G$ は多層論理式である。(4) (1)(2)(3) の規則を有限回くり返えして得られるものは多層論理式である。

2．2　多層論理式の意味

多層論理式 $G(x_0, \cdots, x_n)$ が変数 $x_0, \cdots, x_n$ を自由変数として含み，それらの変数の領域がある集合 D と $Dom\{x_0\} \in \cdots \in Dom\{x_n\} = *_n \cdots *_1 D$ なる関係を満足するとき，

$$(Q_n x_n / *_n \cdots *_1 D)(Q_{n-1} x_{n-1} / x_n) \cdots (Q_0 x_0 / x_1)\, G(x_0, \cdots, x_n)$$

$$= \mathop{\mathbb{W}}_{i_n=1}^{c_n} (Q_{n-1} x_{n-1} / d^n_{i_n}) \cdots (Q_0 x_0 / x_1)\, G(x_0, \cdots, d^n_{i_n})$$

$$\cdots$$

$$= \mathop{\mathbb{W}}_{i_n=1}^{c_n} \cdots \mathop{\mathbb{W}}_{i_0=1}^{c_0} G(d^0_{i_0}, \cdots, d^n_{i_n})$$

ここで，$c_n, \cdots, c_0$ はそれぞれ $Dom\{x_n\}, \cdots, Dom\{x_0\}$ の濃度を示す。また，$1 \leqq i_n \leqq c_n, \cdots, 1 \leqq i_0 \leqq c_0$ なる $i_n, \cdots, i_0$ に対し $d^n_{i_n} \in *_n \cdots *_1 D, \cdots, d^0_{i_0} \in Dom\{x_0\}$ である。記号 $\mathbb{W}$ は $Q = \forall$ のとき $\wedge$ (AND)，$Q = \exists$ のとき $\vee$ (OR) を示す。なお，論理式 G の評価は古典論理に従うものとする。

2．3　多層論理式に関する基本的な定理

次に多層論理と Many Sorted Logic との関連を示す定理を証明する．

<u>定理1</u>　ある集合 D が存在し，変数 $x_0, \cdots, x_n$ の領域が $Dom\{x_0\} \in \cdots \in Dom\{x_n\} = *_n \cdots *_1 D$ なる関係を満足するとす

る。多層論理式 G が変数 x_i ($i=1,\cdots,n$) を自由変数として含まなければ $(\forall x_n / *_n \cdots *_1 D)\cdots(\forall x_0 / x_1)\, G(x_0) = (\forall x_0 / D)\, G(x_0)$ が成り立つ。

証明　(⇒)　定義より

$$(\forall x_n / *_n \cdots *_1 D)\cdots(\forall x_0 / x_1)\, G(x_0) = \bigwedge_{i_n=1}^{c_n} \cdots \bigwedge_{i_0=1}^{c_0} G(d^0_{i_0})$$

G は変数 $x_1 \cdots x_n$ を含まないから $\bigwedge_{i_n=1}^{c_n} \cdots \bigwedge_{i_0=1}^{c_0} G(d^0_{i_0}) = \boldsymbol{T}$ ならば $\bigwedge_{i_0=1}^{c_0} G(d^0_{i_0}) = \boldsymbol{T}$ である。他方，$d^0_{i_0} \in D$ であるからこのことは $(\forall x / D)\, G(x) = \boldsymbol{T}$ であることを意味する。すなわち $(\forall x_n / *_n \cdots *_1 D)\cdots(\forall x_0 / x_1)\, G(x_0) \rightarrow (\forall x / D)\, G(x)$ は妥当な論理式である。

(⇐)　$(\forall x / D)\, G(x) = \boldsymbol{T}$ ならば，任意の $d^1_{i_1} \subseteq D$ に対して $(\forall x_0 / d^1_{i_1})\, G(x_0) = \boldsymbol{T}$ である。一方，$d^1_{i_1} \in *D$ であるから $(\forall x_1 / *D)(\forall x_0 / x_1)\, G(x_0) = \boldsymbol{T}$ である。ただし，x_1 は $*D$ を領域とする変数である。以下同様にして $d^n_{i_n} \subseteq *_{n-1} \cdots *_1 D$ なる任意の $d^n_{i_n}$ に対して $(\forall x_{n-1} / d^n_{i_n})\cdots(\forall x_0 / x_1)\, G(x_0) = \boldsymbol{T}$ である。他方，$d^n_{i_n} \in *_n \cdots *_1 D$ であるから $(\forall x_n / *_n \cdots *_1 D)\cdots(\forall x_0 / x_1)\, G(x_0) = \boldsymbol{T}$ である。すなわち $(\forall x / D)\, G(x) \rightarrow (\forall x_n / *_n \cdots *_1 D)\cdots(\forall x_0 / x_1)\, G(x_0)$ は妥当な論理式である。Q.E.D.

補助定理1　A，B を $A \subset B$ なる任意の集合とする。G が x

を自由変数として含む論理式であるとき，$(\exists x/A)G(x) \rightarrow (\exists x/B)G(x)$は妥当な論理式である。

<u>証明</u>　Many Sorted Logicの定義より $(\exists x/A)G(x) = (\exists x)((x \in A) \wedge G(x))$，$(\exists x/B)G(x) = (\exists x)((x \in B) \wedge G(x))$ である。$(\exists x/A)G(x) = \mathbf{T}$ であるならば $a \in A$ であり，かつ $G(a) = \mathbf{T}$ であるような要素 a が存在する。また，$A \subset B$ であるから $a \in A$ ならば $a \in B$ である。よって $(\exists y)((y \in B) \wedge G(x)) = \mathbf{T}$ である。Q.E.D.

<u>定理2</u>　変数 $x_0, \cdots, x_n$ が $Dom\{x_0\} \in \cdots \in Dom\{x_n\} = *_n \cdots *_1 D$ なる関係を満す変数であり，G が変数 x_i $(i = 1, \cdots n)$ を自由変数として含まない論理式であるならば $(\exists x_n/*_n \cdots *_1 D)\cdots(\exists x_0/x_1)G(x_0) = (\exists x_0/D)G(x_0)$ である。

<u>証明</u>　$(\Rightarrow)$　$(\exists x_n/*_n \cdots *_1 D)\cdots(\exists x_0/x_1)G(x_0) = \mathbf{T}$ であるならば $*_n \cdots *_1 D$ の部分集合であり，かつ $(\exists x/d^n_{i_n})\cdots(\exists x_0/x_1)G(x_0) = \mathbf{T}$ である領域 d^n の集合 $\{d^n_1, \cdots, d^n_{k_n}\}$ $(k_n \leqq C_n$，ただし C_n は $*_n \cdots *_1 D$ の濃度$)$ が存在する。すなわち，$(\exists x_n/*_n \cdots *_1 D)\cdots(\exists x_0/x_1)G(x_0) = \bigvee_{i_n=1}^{k_n}(\exists x_{n-1}/d^n_{i_n})\cdots(\exists x_0/x_1)G(x_0)$ が成り立つ。このような各 $d^n_{i_n}$ に対し上記と同様な操作を繰り返すことにより $(\exists x_n/*_n \cdots *_1 D)\cdots(\exists x_0/x_1)G(x_0) = \bigvee_{i_n=1}^{k_n}\cdots\bigvee_{i_1=1}^{k_1}(\exists x_0/d^1_{i_1})G(x_0)$ を得る。ところが $d^1_{i_1} \subseteq D$ であるから補助定理

1より $\bigvee_{i_n=1}^{k_n}\cdots\bigvee_{i_1=1}^{k_1}(\exists x_0/d_{i_1}^1)G(x_0)\rightarrow(\exists x_0/D)G(x_0)$ は妥当な論理式である。すなわち，$(\exists x_n/*_n\cdots*_1D)\cdots(\exists x_0/x_1)G(x_0)\rightarrow(\exists x_0/D)\ G(x_0)$ は妥当な論理式である。

$(\Leftarrow)$　$(\exists x_0/D)G(x_0)=\mathbf{T}$ であるならば，$e\in D$ であり $G(e)=\mathbf{T}$ であるような e が存在する。e を要素として含む D の全ての集合を $E_1,\cdots,E_{k_1}$ とすると $(\exists x_0/D)G(x_0)=\bigwedge_{i_1=1}^{k_1}(\exists x_0/E_{i_1})G(x_0)$ である。一方，任意の論理式 $A_i\ (i=1\cdots k)$ に対して $\bigwedge_{i=1}^{k}A_i\rightarrow\bigvee_{i=1}^{k}A_i$ は妥当な論理式であるから，$\bigwedge_{i_1=1}^{k_1}(\exists x_0/E_{i_1})G(x_0)\rightarrow\bigvee_{i_1=1}^{k_1}(\exists x_0/E_{i_1})G(x_0)$ は妥当な論理式である。さらに，多層論理式の定義より $\bigvee_{i_1=1}^{k_1}(\exists x_0/E_{i_1})G(x_0)=(\exists x_1/*D)(\exists x_0/x_1)G(x_0)$ である。すなわち，$(\exists x_0/D)G(x_0)\rightarrow(\exists x_1/*D)(\exists x_0/x_1)G(x_0)$ は妥当な論理式である。この操作をさらに $n-1$ 回繰り返すことにより $(\exists x_0/D)G(x_0)\rightarrow(\exists x_n/*_n\cdots*_1D)\cdots(\exists x_0/x_1)G(x_0)$ が妥当な論理式であることを得る。

3．多層論理式によるリレーショナル・データベースに関する知識と質問の記述，並びに推論処理.

3．1　多層論理式のアトムの分類

アトムの定義は2．1で与えたが，以後の説明の都合上アトムを以下に示す4種類に分類する。

1） FGETアトム：これはデータベースのリレーションを表現するモデルアトムである．これを（FGET，t_0，t_1，…t_n）（$n \geqq 1$）と表現する。ここでt_0はn項リレーションの名前であり，t_i（$i=1 \cdots n$）はリレーションの第iカラム目を示す項である。

2） 判別型アトム（DTA）：これは引数に与えられた値がアトム名で示されている関係を満すかどうかを判定するアトムである。例えば EQ（=），GT（>），SSET（⊇）．

3） 演算型アトム（MTA）：多層論理式では関数を表現するためにその関数に相当する表現述語（Representation Predicate）を用いる．この表現述語を演算型アトムと呼ぶ。

4） ユーザ・アトム：ユーザが問題を記述するために定義するアトム．このアトムはFGETアトム，判別型アトム，演算型アトム，及びすでに定義されているユーザ・アトムを用いて定義される．

ユーザ・アトムを評価するためには，これを含む論理式を用いて展開しなければならないのに対し，FGET，判別型，演算型の各アトムはデータベース又はシステムが備えているライブラリを用いて評価する。この意味で，ユーザ・アトムを間接評価アトム（IPA），それ以外のアトムを直接評価アトム（DPA）と呼ぶことがある．

3.2 多層論理式によるリレーショナル・データベースに関する知識と質問の記述

本節では多層論理式によるリレーショナル・データベースの検索に関する知識と質問の記述方法を簡単な例を用いて論ずる。以下で用いるリレーションは

CLASS (STUD-ID , DEPT , COURSE)

STUDY (STUD-ID , SUBJ-ID , MARK)

SUBJECT (SUBJ-ID , TITLE , DEPT)

である。ソート（属性）STUD-ID，DEPT，SUBJ-ID，MARK，TITLEは，それぞれ学生識別子，学部，学科，科目識別子，得点，科目名を示す。CLASSは，学生が属している学部，学科を表わすリレーションである。STUDYは学生が履習した科目と得点を表わすリレーションである。SUBJECTは科目とその題名と学部の関係を表わしている。このデータベースに対する質問の例を次に示す。

QU1. 工学部の学生が履習している教養学部の科目の科目ごとの平均点を科目名とともに求めよ。

QU2. 工学部・航空学科の全ての学生が履習している科目の科目ごとの平均点を求む。

QU3. 工学部・情報処理学科の2人以上の学生が履習している科目の科目ごとの平均点を求む。

QU4. 工学部・航空学科のどの学生も履習していない科目の科目数を求む。

QU5. 工学部・航空科又は情報工学科の学生が履習している科目の科目ごとの平均点を求む。

QU1 ～ QU5 は，それぞれ General quantification, Universal quantification, Numerical quantification, Negation, Disjunctive な記述を含む質問の例である。これらの質問を記述するために，データベースのリレーションに関する知識を記述する。

```
K1.    (∀S/*STUD-ID)(∀D/*DEPT)(∀C/*COURSE)(∀s/S)(∀d/D)(∀c/C)
       [ ~(FGET, CLASS, s, d, c) v (CLA, s ,d, c) ].

K2.    (∀S/*STUD-ID)(∀J/*SUBJ-ID)(∀M/*MARK)(∀s/S)(∀j/J)(∀m/M)
       [ ~(FGET, STUDY, s, j, m) v (STU, s, j, m ) ].

K3.    (∀S/*STUD-ID)(∀T/*TITLE)(∀D/*DEPT)(∀s/S)(∀t/T)(∀d/D)
       [ ~(FGET, SUBJECT, s, t, d ) v (SUBJ, s, t, d ) ]
```

これらの知識を用いて QU1 ～ QU5を記述することができる。例えば QU1は

```
Q1'.   (∃c/COURSE)(∃s/STUD-ID)(∃t^/TITLE)(∀j^/SUBJ-ID)
       (∃M/*mark)(∀m/M)(∃a^/MARK)
       [ (CLA, s, 'E'/DEPT, c) & (STU, s, j, m)
        &(SUBJ, j, t, 'L'/DEPT ) & (AVERAGE, M, a) ] ?
```

のように書ける。しかし，上記の質問を記述する限り K1，K2 の代わりに次の知識を用いた方がわかりやすい。

```
K4.     (∀D/*DEPT)(∀C/*COURSE)(∀S/*STUD-ID)(∀J/*SUBJ-ID)(∀M/*MARK)
        (∀d/D)(∀c/C)(∀s/S)(∀j/J)(∀m/M)
        [ ~(FGET, CLASS, s, d, c) v ~(FGET, STUDY, s, j, m)
         v(STU-STU, d, c, s, j, m) ].
```

アトム STU-STU はd学部，c学科の学生sが科目jで評価mであることを表わしている。次に示すQ1～Q5はK3とK4を用いて上記の質問QU1～QU5を表現したものである。

```
Q1.     (∃c/COURSE)(∃s/STUD-ID)(∃t^/TITLE)(∀j^/SUBJ-ID)
        (∃M/*MARK)(∀m/M)(∃a^/MARK)
        [ (STU-STU, 'E'/DEPT, c, s, j, m )
         &(SUBJ, j, t, 'L'/DEPT)
         &(AVERAGE, M, a) ] ?

Q2.     (∃j/SUBJ-ID)(∃M/*MARK)(∀m/M)(∃a/MARK)(∀s/STUD-ID)
        [ (STU-STU, 'E'/DEPT, 'AER'/COURSE, s, j, m )
         &(AVERAGE, M, a ) ] ?

Q3.     (∀j^/SUBJECT)(∃S/*STUD-ID)(∃M/*MARK)
        (∀s/S)(∃ns/INTG)(∀m/M)(∃a^/MARK)
        [ (STU-STU, 'E'/DEPT, 'INF'/COURSE, s, j, m )
         &(AVERAGE, M, a )
         &(COUNT, S, ns )
         &(GT, ns, #2/INTG ) ] ?

Q4.     (∃c/COURSE)(∃s/STUD-ID)(∃m/MARK)(∃J/*SUBJ-ID)(∀j/J)
        (∃nj^/INTG)
        [ (STU-STU, 'E'/DEPT, c, s, j, m )
         &(NE, c, 'AER'/COURSE)
         &(COUNT, J, nj ) ] ?

Q5.     (∃c/COURSE)(∃s/STUD-ID)(∀j^/SUBJ-ID)
        (∃M/*MARK)(∀m/M)(∃a^/MARK)
        [ (STU-STU, 'E'/DEPT, c, s, j, m )
         &( (EQ, 'AER'/COURSE, c )
           v(EQ, 'INF'/COURSE, c ) )
         &(AVERAGE, M, a ) ] ?
```

これらの記述にはデータベースのリレーションに関する物理的な情報は含まれていない。例えば，ユーザ・アトムSTU－

STU は STU-STU（DEPT，COURSE，STUD-ID，SUBJ-ID，MARK）なるリレーションが存在するかのように用いられているが，このリレーションそのものがデータベースに定義されている必要はない。論理的なリレーションと物理的なリレーションとの関連は K1～K4 のように FGET アトムを含む論理式によって定義される。

3．3 推論処理

ユーザ・アトムを含む質問をデータベースとシステムが備えているライブラリを用いて直接評価することはできない。そのために，ユーザ・アトムをその定義を用いて展開する必要がある。この処理を推論処理と呼ぶ。多層論理における推論処理の原理は一階述語論理における導出原理と同様，質問内に含まれるユーザ・アトムを，それを含意するアトムを含む論理式によって置換し新たな質問を作り出す。この処理は質問を構成するアトムが全て DPA になるまで続けられる。推論処理アルゴリズムの詳細については他稿に譲り[8,10,11]，ここでは推論処理によってユーザ・アトムを含む論理式が DPA のみから構成されている論理式に変換されてゆく様子を Q1 を例題として述べよう．

推論処理は反駁証明法によって行なう。したがって Q1 の否定形を作る。

```
~Q1.    (∀c/COURSE)(∀s/STUD-ID)(∀t^/TITLE)(∃j^/SUBJ-ID)
        (∀M/*MARK)(∃m/M)(∀a^/MARK)
        [ ~(STU-STU, 'E'/DEPT, c, s, j, m )
         v~(SUBJ, j, t, 'L'/DEPT)
         v~(AVERAGE, M, a ) ] ?
```

次に，質問内の ~(STU-STU, 'E'/DEPT, $\forall c'$, $\forall s'$, $\exists j'$, $\exists m'$) とK4内の (STU-STU, $\forall \alpha$, $\forall c$, $\forall s$, $\forall j$, $\forall m$) との同一化を行なう。含意条件 $\alpha \leftarrow$ 'E'/DEPT, $Dom\{c\} \wedge Dom\{c'\}$ = COURSE $\neq \phi$, $Dom\{s\} \wedge Dom\{s'\}$ = STUD-ID $\neq \phi$, $Dom\{j\} \supseteq Dom\{j'\}$, $Dom\{m\} \supseteq Dom\{m'\}$ を満しているので，質問Q1 は K4 によって次の論理式に変換される。

```
~Q1 ⊕ K4.

(∀c/COURSE)(∀s/STUD-ID)(∀t^/TITLE)(∃j^/SUBJ-ID)
(∀M/*MARK)(∃m/M)(∀a^/MARK)
[ ~(FGET, CLASS, s, 'E'/DEPT, c) v ~(FGET, STUDY,s, t, m)
 v~(SUBJ, j, t, 'L'/DEPT)
 v~(AVERAGE, M, a) ] ?
```

同様にして次の論理式を得る。

```
~Q1 ⊕ K4 ⊕ K3.

(∀c/COURSE)(∀s/STUD-ID)(∀t^/TITLE)(∃j^/SUBJ-ID)
(∀M/*MARK)(∃m/M)(∀a^/MARK)
[ ~(FGET, CLASS, s, 'E'/DEPT, c) v ~(FGET, STUDY, s, j, m)
 v~(FGET, SUBJECT, j, t, 'L'/DEPT)
 v~(AVERAGE, M, a ) ] ?
```

この論理式に含まれるアトムは全てDPAであるために推論処理を終了する。

4. 多層論理式からのデータベース検索手続きの生成

4.1 多層論理式のグラフによる表現

推論処理を終了した多層論理式の上記の表現は，それを評価するために必要な情報の全てを表現していない。推論処理の段階で導入される変数の順序は一般に半順序をなす。この情報は多層論理式を評価する手続きを生成するためには不可欠なものであるが，上記の論理式には表現されていない。これらの情報を表現するために，多層論理式を以下に定義するCAVグラフによって表現する。CAVグラフは多層論理式の計算機の内部表現を抽象化したものである。

定義　CAVグラフとは有向グラフ $G=(V, E)$ である。ここで，ノードの集合Vは $V=V_A \cup V_c \cup V_V$，ただし $V_A \cap V_c = \phi$，$V_c \cap V_c = \phi$，$V_V \cap V_A = \phi$ である。また，アークの集合Eは $E = E_{AC} \cup E_{AV} \cup E_V$，ただし $E_{AC} \cap E_{AV} = \phi$，$E_{AV} \cap E_V = \phi$ $E_V \cap E_{AC} = \phi$ である。また，$E_{AC} = \{(u, v) \mid u \in V_A \text{ and } v \in V_c\}$，$E_{AV} = \{(u, v) \mid (u \in V_A \text{ and } v \in V_V) \text{ XOR } (u \in V_V \text{ and } v \in V_A)\}$，$E_V = \{(u, v) \mid u \in V_V \text{ and } v \in V_V\}$ である。

V_A，V_c，V_V によって，それぞれ論理式内のアトム，定数，変数の集合を表わす。$(u, v) \in E_{Ac}$ はアトムuの引数に定数vが含まれていることを示すアークであり，$(u, v) \in E_{AV}$

は，$u \in V_A$，$v \in V_V$であるときアトムuを評価することによって変数vの値が定まることを示し，$u \in V_V$，$v \in V_A$であるときアトムvを評価するためには変数uの値が定まらなければならないことを示す。$(u, v) \in E_V$は変数uがvに先行していることを示すアークである。このCAVグラフを用いて3.3で得られた論理式 $\sim Q1 \oplus K4 \oplus K3$を表現したものを図1に示した。

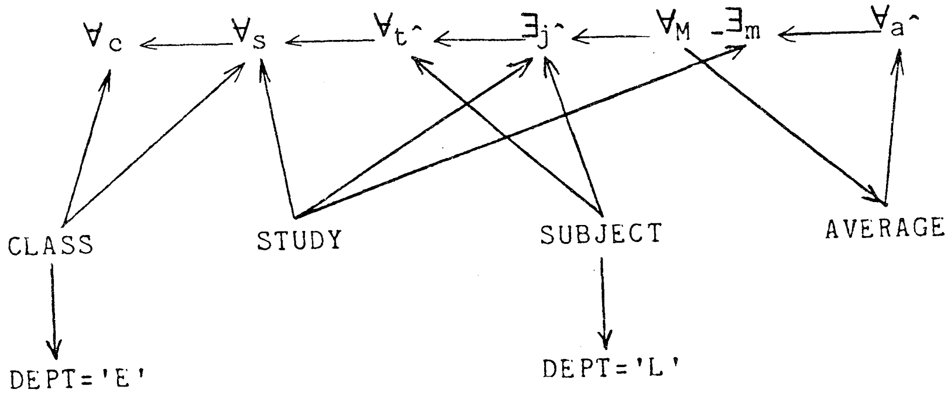

図1．CAVグラフによる多層論理式の表現の例。

4．2 多層論理式を評価する手続きを生成するアルゴリズム

述語論理式からデータベースの検索手順を生成するアルゴリズムは Reiter [1978]，Chang [1980] などによって論じられている。しかし，彼等が対象とした論理式は Many Sorted Logic の範囲であり，集合変数を陽に含む多層論理式に対して彼らが示したアルゴリズムをそのまま適用することはできない。多層論理式と Many Sorted Logic の評価で異なるのは

多層論理式を評価するためには評価のある段階で集合変数の値を確定する必要があることである。Many Sorted Logic の範囲では変数の値は全て要素定数であった。したがって推論処理または評価の過程のいずれかにおいて変数の値が一度定まれば，その値は以後の処理において変更されることはない。これに対して多層論理式では集合を値とする変数を含んでいる。集合変数の値を定める条件は，その集合変数に先行する∀限量された変数の有無によって異なる．

1）注目している集合変数に先行する∀限量された変数が無いとき，集合変数を制限する全ての条件を評価した段階でその値が定まる。

2）注目している集合変数に先行している∀限量された変数があるときは，集合変数に先行する∀限量されている変数の値が定まったとき，これらの変数の値によって集合の値を集約したそれぞれの集合を，この集合変数の値とする。

2）の条件は，注目している集合変数がこれに先行する∀変数と関数関係にあるために，上記の処理によって集合変数の値が定まる。一階述語論理のスコーレム標準形では，この関数関係をスコーレム関数を用いて表現することができる。多層論理では∃限量された変数を陽に含めているために，評価

手順生成アルゴリズムはスコーレム関数に相当するものを自動的に生成しなければならない。なお 2) における処理は，Chamberlin, etc. [1976] の SEQUEL2 における GROUP BY に相当するものである。

次に DPA のみから構成されている多層論理式を評価する手続きを生成するアルゴリズムを示す。

Step 1. FGET アトムを全て評価する。必要に応じて Restriction Join, Intersection, Union, Difference, Product を行う。

Step 2. 評価可能な DTA を全て評価する。続いて，評価可能な MTA を全て評価する.

Step 3. 先行する $\forall$ 限量された変数の値が全て定まった集合変数の値を決める。

Step 4. $\forall$ 限量されている変数が先行している全ての集合変数の値が定まったなら Step 5，そうでなければ Step 2 へ行く。

Step 5. 値が定まっている変数で，答えとして要求されていない変数に対し，$\bar{Q}' = \exists$ なら Division，$\bar{Q}' = \forall$ なら Projection を行なう。

Step 6. 全ての限量子の評価が終了したならば STOP，そうでなければ Step 7 へ行く。

Step 7. 評価可能な DTA を全て評価し，続いて評価可能な MTA

を全て評価する。Step 5 へ行く.

例として，このアルゴリズムを3.3で得られた多層論理式~Q ⊕K4 ⊕K3 に適用することによって，この論理式を評価する手続きが生成されてゆく様子をCAVグラフを用いて説明する。

まず，論理式中のFGET アトムを全て評価し，定数を含むアトムに対してはその定数に関するRestrictionを行なう。この処理によって図1のCAVグラフは図2のように変わる。図

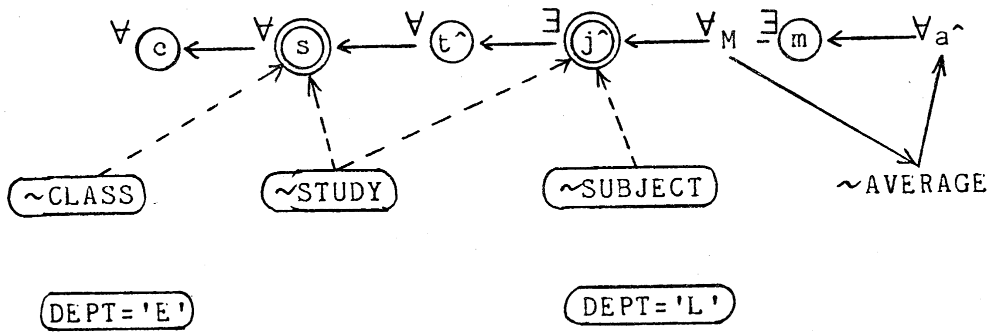

図2. Restriction の処理が終了した時点でのCAVグラフ。

2ではノードvの評価が終了したことを⓪で表現している。ノード s，j ∈ Vv は，それぞれ（FGET, CLASS, ---）と（FGET, STUDY, ---），（FGET, STUDY, ---）と（FGET, STUDENT, ---）が評価されることによって1度づつ評価されるために◎の印が付いている。これはリレーション CLASS と STUDY を s で Join し，さらにこの Join によって得られた結果のリレーションと SUBJECT とを j で Join すべきであることを意味している。これらの処理を終了した時点でのCAVグラフを

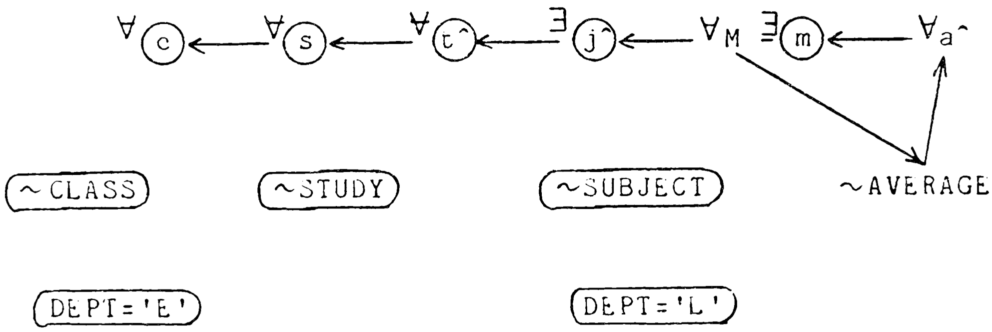

図3．JOIN の処理が終了した時点でのCAVグラフ．

図3に示す。図3で集合変数Mに先行している∀限量された変数j（図3では否定形を表現しているので∃限量されている）の値が定まっているのでjの各値に対してMの値を定める。これによって演算型アトムAVERAGEが評価可能になるのでこれを評価する。続いて限量子を評価することによって全ての処理を終える。このときのCAVグラフの状態を図4に示す。また生成された手続きを図5に示した。

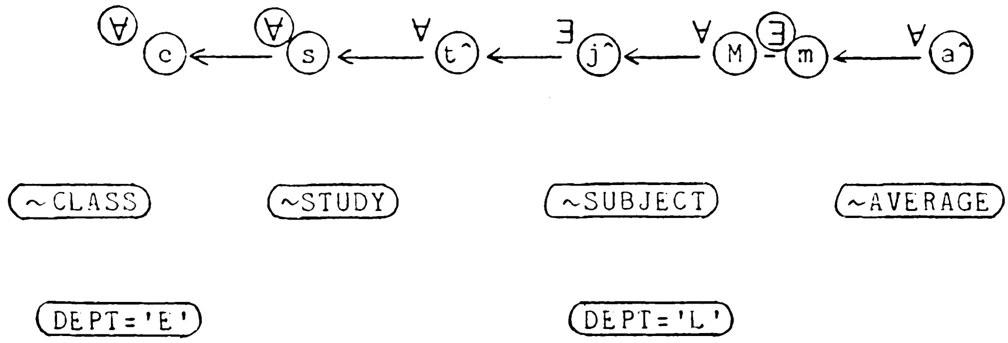

図4．生成アルゴリズムが停止した時点のCAVグラフ。

5．おわりに　本稿では多層論理と推論処理，並びにデータ

ベース又はシステムが備えているライブラリ・プログラムを用いることによって評価されるアトムから構成されている多層論理式から，それを評価する手続きを生成するアルゴリズムについて論じた。多層論理式は値として要素をとる変数のみならず集合又は巾集合などを値とする変数を含む知識を統一的に記述し，しかも推論処理が行なえる形式言語である．ここでは，多層論理式の定義を与え基本的な定理を証明した。これによって多層論理とMany Sorted Logicとの相違点が明らかになった。また，簡単なデータベースを用いて多層論理式によるリレーショナル・データベースの検索に関する知識と質問の記述方法を示した。さらに推論処理によって質問がデータベースと基本的なライブラリ・プログラムを用いることによって評価できる論理式に変換されることを示した。多層論理式の表現では推論の途中に導入される変数の順序情報を表現するために不十分であるためにCAVグラフを定義し，これによって多層論理式を評価する手続きを生成するアルゴリズムを説明した。

```
FGET     CLASS
FGET     STUDY
FGET     SUBJECT
REST     CLASS[DEPT='E']
REST     SUBJECT[DEPT='L']
JOIN     R1= CLASS[s]STUDY
JOIN     R2=.R1[j]SUBJECT
DETERM   M  for each j.
AVERAGE
DIV      m/M
PROJ     s
PROJ     c
```

図5．生成された手続き。

今後の研究課題としては，生成された手続きを最適化する

ことなどが残されている。

REFERENCE

1). Chamberlin D.D., etc. [1976] SEQUEL 2 : a unifiedapproach to data definition, manipulation and control, IBM J. Res. Develop. 20-6 pp560-575.

2) Chang,C.L. [1980] On evaluation of queres containing derived relations in a relational database. Advances in Data Base Theory - Vol 1, Plenum Press,New York.

3) Clark,K.L. [1978] Negation as Failure, In Logic and Data Bases ,Plenum Press, New York, 1978, pp293-322.

4) Kellog,C., Klahr,P. and Travis,L. [1978] Deductive Planning and Pathfinding for Relational Data Bases, In Logic and Data Bases, Plenum Press, New York, 1978, pp179-200.

5) Kowalski,J. [1979] Algorithm = Logic + Control, c.ACM 22.7, pp424-436.

6) Minker,J. [1978] An experimental Relational Data Base System Based on Logic, In Logic and Data Bases, Plenum Press, New York 1978, pp107-147.

7) Reiter,R. [1978] Deductive Question-Answering on Relational Data Bases, In Logic and Data Bases, Plenum Press, New York, 1978, pp149-177.

8) Ohsuga,S. [1979] Toward Intelligent Interactive Systems. Proc.The IFIP W.G 5.2.

9. 宇田川，大須賀　：述語論理に基づく知識システムのマンマシン・インターフェースについて、情報システム研究会, 1980.

10. 宇田川，大須賀　：述語論理に基づいた知識システムの実現と今後の発展　総合研究「知識工学の基礎とその応用」第4回　1981.

11. 大須賀　：知識表現のための多層論理．人工知能と対話技法研究会，17-6，1980.